"Express Mail" Mailing Label No. **EL960827955US**

PATENT APPLICATION
ATTORNEY DOCKET NO. SUN-P9311-SPL

METHOD AND APPARATUS FOR PREVENTING SPANNING TREE LOOPS DURING TRAFFIC OVERLOAD CONDITIONS

Inventor: Radia J. Perlman

BACKGROUND

Field of the Invention

[0001] The present invention relates to the design of computer networks. More specifically, the present invention relates to a method and an apparatus that prevents loops from occurring when spanning tree configuration messages are lost while executing a spanning tree protocol across bridges in a network.

Related Art

[0002] Computer networks are frequently coupled together through transparent bridges. The most basic form of transparent bridge is one that attaches

to two or more local area networks (LANs) (each attachment to a bridge is referred to as a "port"). Such a bridge listens promiscuously to every packet transmitted and stores each received packet until it can be transmitted on the LANs other than the one on which it was received.

[0003] The transparent bridge was developed to allow stations that were designed to operate on only a single LAN to work in a multi-LAN environment. The stations expect to transmit a packet, exactly as they would in a single-LAN environment, and have the packet delivered. The bridge must therefore transmit the packet exactly as received. If the bridge modified the packet in any way--for example, by overwriting the source address portion of the header with its own address--then protocols on stations might not work properly.

[0004] Note that bridges can potentially cause a packet to loop, which can cause the packet to replicate exponentially. This replication can increase congestion on the network to the point where the network stops functioning.

[0005] This looping problem is commonly dealt with by using a spanning tree protocol defined in Institute of Electrical and Electronics Engineers (IEEE) standard 802.1D. This spanning tree protocol operates by having bridges dynamically discover a subset of the network topology that is loop-free (a tree) and yet has enough connectivity so that, where physically possible, there is a path between every pair of LANs (the tree is *spanning*).

[0006] The basic idea behind the spanning tree protocol is that bridges periodically transmit special configuration messages to each other that allow them to calculate a spanning tree. Referring to FIG. 2, these configuration messages contain enough information to allow bridges to do the following. (1) Elect a single bridge among all bridges on all LANs, to be the *root bridge* (step 202). (2) Calculate the distance of shortest path from themselves to the root bridge (step 204). (3) Elect a designated bridge on each LAN from the bridges residing

on that LAN (step 206), wherein the elected bridge is the one closest to the root bridge and will forward packets to the root bridge. (4) Choose a port for each bridge that gives the best path to the root bridge (step 208). (5) Select ports on each bridge to be included in the spanning tree (step 210). (6) Place selected ports into a forwarding state in which messages are forwarded to and from the port (step 212). (7) Place other ports into a backup state, in which messages are not forwarded to or from the port (step 214).

[0007] This protocol can be summarized in the following poem entitled "Algorhyme" by Radia Perlman, the inventor of the present invention.

I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop free connectivity.
A tree that must be sure to span
So packets can reach every LAN.
First, the root must be selected.
By ID, it is elected.
Least-cost paths from root are traced.
In the tree, these paths are placed.
A mesh is made by folks like me,
Then bridges find a spanning tree.

[0008] It is important to engineer a bridge with sufficient CPU power so that if the network becomes congested, the spanning tree protocol will operate properly. Otherwise, the network becoming temporarily congested might cause configuration messages to become lost, which can cause the spanning tree protocol to incorrectly turn extra bridge ports on. This can cause loops, which can dramatically increase the amount of congestion to such a point that the spanning tree protocol never recovers.

[0009] Unfortunately, the IEEE 802.1D standard does not specify a performance requirement, and as a result, some of the bridge hardware that is

presently deployed is not capable of processing spanning tree configuration messages during worst-case traffic. Consequently, messages can be lost and loops can be created.

[0010] Hence, what is needed is a method and an apparatus that prevents loops from occurring when spanning tree configuration messages are lost.

SUMMARY

[0011] One embodiment of the present invention provides a system that prevents loops from occurring when spanning tree configuration messages are lost while executing a spanning tree protocol on bridges in a network. During operation, the system executes the spanning tree protocol on a bridge. This spanning tree protocol configures each port coupled to the bridge into either a forwarding state, in which messages are forwarded to and from the port, or a backup state, in which messages are not forwarded to or from the port. The system also monitors ports coupled to the bridge to determine when messages are lost by the ports. If one or more messages are lost on a port, the system refrains from forwarding messages to or from the port until no messages are lost by the port for an amount of time.

[0012] In a variation on this embodiment, the amount of time is greater than a time interval provided by bridges between consecutive spanning tree configuration messages.

[0013] In a variation on this embodiment, monitoring ports coupled to the bridge involves communicating with hardware associated with the ports to determine if messages have been lost by the ports.

[0014] In a variation on this embodiment, executing the spanning tree protocol involves placing ports coupled to the bridge into either the forwarding state or the backup state in a manner that ensures that messages are forwarded

without cycling across a spanning tree that couples together bridges in the network.

[0015] In a variation on this embodiment, executing the spanning tree protocol involves: electing a single bridge among all bridges on all links on the network to be a root bridge; calculating the distance of the shortest path from each node to the root bridge; electing a designated bridge for each link from all bridges on the link, wherein the designated bridge is closest to the root bridge and will forward packets from the link to the root bridge; choosing a root port for each bridge that provides the best path to the root bridge; selecting ports on each bridge to be included in the spanning tree, wherein the selected ports include the root port and any ports coupled to links upon which the bridge serves as the designated bridge; placing selected ports into the forwarding state; and placing all other ports into the backup state.

[0016] In a variation on this embodiment, the spanning tree protocol generally operates in accordance with Institute of Electrical and Electronics Engineers (IEEE) standard 802.1D.

[0017] In a variation on this embodiment, the links are Local Area Networks (LANs).

BRIEF DESCRIPTION OF THE FIGURES

[0018] FIG. 1 illustrates an exemplary network with bridges in accordance with an embodiment of the present invention.

[0019] FIG. 2 presents a flow chart illustrating a spanning tree protocol.

[0020] FIG. 3 presents a flow chart illustrating how the spanning tree protocol deals with lost messages in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

[0021] The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

[0022] The data structures and code described in this detailed description are typically stored on a computer readable storage medium, which may be any device or medium that can store code and/or data for use by a computer system. This includes, but is not limited to, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs) and DVDs (digital versatile discs or digital video discs), and computer instruction signals embodied in a transmission medium (with or without a carrier wave upon which the signals are modulated). For example, the transmission medium may include a communications network, such as the Internet.

The Network

[0023] FIG. 1 illustrates an exemplary network 100 with bridges in accordance with an embodiment of the present invention. As is illustrated in FIG. 1, network 100 includes a number of links 106-110. In one embodiment of the present invention, links 106-110 are local area networks (LANs), such as Ethernet-based networks, that couple together local computing nodes (stations). More specifically, in FIG. 1, link 106 couples together nodes 112-114 and bridge

102; link 107 couples together nodes 115-116 and bridges 102-103; link 108 couples together nodes 119-120 and bridges 103-104, link 109 couples together nodes 117-118 and bridges 103-104; and link 110 couples together nodes 121-123 and bridge 104.

[0024] Note that bridges 102-104 are designed to transparently couple together links 106-110 so that they appear to be part of a single combined network.

Spanning Tree Protocol

[0025] The spanning tree protocol generally operates as described above with reference to FIG. 2. However, in some cases network congestion can cause spanning tree configuration messages to be lost, which can cause the spanning tree protocol to incorrectly turn extra bridge ports on. This can possibly cause loops, which can dramatically increase the amount of congestion to a point that the spanning tree protocol never recovers.

[0026] FIG. 3 presents a flow chart illustrating how the spanning tree protocol deals with lost messages to prevent the occurrence of such loops in accordance with an embodiment of the present invention. The system generally executes a spanning tree protocol as described above with reference to FIG. 2 (step 302). At the same time, the system monitors ports coupled to the bridge (step 304). During this monitoring process, the system determines if messages have been "lost" by any ports. Note that messages are "lost" on a port when the system is not able to send or receive one or more messages through the port (step 306).

[0027] It is implementation dependent how a bridge knows that it is not keeping up with traffic. In one embodiment of the present invention, the NIC card informs a driver, which increments counters indicating lost incoming messages

(for example, when an incoming message is lost due to a buffer overrun condition). These counters are available for inspection by the upper layers.

[0028] Another case in which spanning tree meltdowns occur is when a bridge is not capable of transmitting its spanning tree messages. A typical case in which this occurs is where there is a link which is configured as half duplex in one direction and full in the other. In this case, if the full duplex side has sufficient traffic, the half duplex side will not be able to transmit. (Note that the system may try to send a message one or more times before giving up.) This situation will only be detectable by the half duplex side. If that bridge (the one that believes it is half duplex on that link) believes it should be designated bridge on that link, and it cannot transmit its spanning tree messages, then as in the previous scenario, it should continue doing its best to run the spanning tree algorithm itself, but it should not forward data traffic to and from the link.

[0029] If at step 306 the system determines no messages have been lost, the system returns to step 302 to continue executing the spanning tree protocol.

[0030] Otherwise, if a message has been lost by a port, before forwarding messages to or from the port, the system waits until no messages are lost by the port for a sufficient amount of time to ensure that subsequent spanning tree configuration messages are received on the port (step 308). This ensures that the port will not erroneously forward messages as the result of spanning tree configuration messages being lost. This reduces the likelihood that loops will be erroneously generated by the spanning tree protocol.

[0031] Note that if messages are lost on a port that is in forwarding state, the associated bridge may not know that it should have changed the port to the backup state because another bridge is more qualified to be the designated bridge. On the other hand, if messages are lost on a port that is in backup state, the bridge may assume that it is the designated bridge for the port (or possibly the root

bridge) and may erroneously change the port to forwarding state, when there actually exists a more qualified bridge.

[0032] The foregoing descriptions of embodiments of the present invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended claims.